AN
INTERNSHIP REPORT

ON

TOLLGATE MANAGEMENT SYSTEM PROJECT REPORT

BY

KAMAL ACHARYA

(Tribhuvan University)

Date: 2023/12/07

1. INTRODUCTION

Toll Gate Management application have been of great assistance in lessening the over congestion that has become a part of the metropolitan cities these days. It is one of the uncomplicated ways to manage the great run of traffic The travelers passing through this mode of transport, carried by their transport that allows them to be aware of the account of money that has been paid and the money left in the tag. It relieves the traveler of the burden of waiting in the queue to make the toll payment, which decreases the fuel-consumption and also taking cash with them can be avoided. Our application avoid this type of problems. user get gate pass from online so user don't need to wait in tollgate.

The Tollgate management is designed from a user point of view. The user friendly design helps the users in accomplishing their task with ease. Attempts have been made to keep the design simple and understandable. The screens were designed in XML and the business logic was written in Java. The total lines of code written in this application are Java, xml .

1.1 Android:

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

Operating Systems have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in then to Blackberry OS and Android.

One of the most widely used mobile OS these days is **ANDROID**. **Android** does a software bunch comprise not only operating system but also middleware and key applications. Android Inc was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. After original release there have been number of updates in the original version of Android.

Android's releases are named after sweets or dessert items, except for the first and second releases:

1.0 – (No codename) (API Level 1)

1.1 – (Internally known as "Petit Four") (API Level 2)

1.5 – Cupcake: (API Level 3)

1.6 – Donut: (API Level 4)

2.0 – Eclair: (API Level 5)

2.0.1 – Eclair: (API Level 6)

2.1 – Eclair: (API Level 7)

2.2.x – Froyo: (for "Frozen Yogurt"): (API Level 8)

2.3 – Gingerbread: (minor UI tweak): (API Level 9)

2.3.3 – Gingerbread: (API Level 10)

3.0 – Honeycomb: (major UI revamp): (API Level 11)

3.1 – Honeycomb: (API Level 12)

3.2 – Honeycomb: (API Level 13)

4.0 – Ice Cream Sandwich: (minor UI tweak): (API Level 14)

4.0.3 – Ice Cream Sandwich: (API Level 15)

4.1 – Jelly Bean: (API Level 16)

4.2 – Jelly Bean: (API Level 17)

4.3 – Jelly Bean: (API Level 18)

4.4.4 – KitKat: (API Level 19)

4.4W – KitKat Watch: (API Level 20)[11]

5.0, 5.0.1, 5.0.2 – Lollipop: (major UI revamp): (API Level 21)

5.1, 5.1.1 – Lollipop: (API Level 22)

6.0 & 6.0.1 – Marshmallow: (API Level 23)

7.0 – Nougat: (API Level 24)[12]

7.1, 7.1.1 – Nougat: (API Level 25)[13]

8.0 – Oreo: (minor UI tweak): (API Level 26)

8.1 – Oreo: (API Level 27)[14]

9.0 – Pie: (API Level 28)

Android One is a software experience that run on the unmodified Android operating system, which closely resembled those which running on Pixel devices or previously, the Google Nexus program. Unlike most of the "stock" Android that running on the market, Android One UI closely resemble of the Pixel UI, due to Android One program are software experience that develop by Google and distribute to partner who signup for the program, such as Nokia and Xiaomi. Thus, the overall UI are intended to be clean as possible (Intended by Google), while OEM partners may tweak or add additional apps such as camera to the firmware, otherwise most of the apps will be handle by Google proprietary apps. The update was handle by Google and will be internally tested by OEM before distribute via OTA update to the end users.

Modules:

The Tollgate management is designed from a user point of view. The user friendly design helps the users in accomplishing their task with ease. Attempts have been made to keep the design simple and understandable. The screens were designed in XML and the business logic was written in Java. The total lines of code written in this application are Java, xml .

Login:

This is the first module in our system .it give authentication to the system. The Login Form module presents site visitors with a form with username and password fields. If the user enters a valid username/password combination they will be granted access to additional resources on your project.

Vehicle category:

This is the second module in our project. We can divide vehicle in four category .two wheeler, lmv(car) ,medium(bus),heavy(truck)… The tollgate pass amount is varied based on vehicle type. The admin already set amount for the vehicle. Suppose user want to get pass for the two wheeler he must wants to click bike button .

Pass type:

This module used to user select pass type to the user. The pass type divide into three categories. Daily pass , month, year .the user click radio button what type of pass he want .and also user give her vehicle number. the pass contain staring date and end date of the system.

Payment pass:

Finally user get pass from system.it is stored on your android emulator..
It contain all information.

2. REQUIREMENT ANALYSIS

2.1 Software and Hardware Requirement

In order to install the software your system must the following specification

2.1.1 Hardware Requirement:

Processor	**:** Pentium IV or higher
RAM	**:** 256 MB
Space on Hard Disk	**:** Minimum 512MB
Device	**:** Mobile Phone,Emulator
Minimum space to execute	**:** 5.0MB

2.1.2 Software Requirement:

Operating System	**:** Windows 10
IDE	**:** Android Studio 3.1
Coding Language	**:** JAVA, XML
Database	**:** SQLite

3. SOFTWARE REQUIREMENTSPECIFICATION

3.1 Software Requirement Specification

An **SRS** is basically an organization understanding of customer or potential client system requirement and dependencies at a particular point in time prior to actual design or development work.

The **SRS** document itself states in precise and explicit language those function And capabilities a software system must provide, as well as state any required constraints by which the system abide. The **SRS** also function as a blueprint for complementing a project with a little cost growth as possible. The **SRS** is often referred to as the parent document because all subsequent project management document, such as design

Specification, statement of work, software architecture specification, testing and validation plans, and documentation plans are related to it.

It's important to note that as SRS contains functional and non-functional requirements only;it doesn't offer design suggestion, possible solution to technology or business issues,or any other information other than what the development team understand the customer's system requirement to be.

Major Goals:

It provide feedback to the customer. An **SRS** is the customer assurance that the development organization understand the issues to problem to be solved and the software behavior necessary to address those problems. Therefore the SRS should be written in natural language, in an unambiguous manner that may also include charts, tables, dfd, decision tables, and so on.

It decomposes the problem into component parts. The simple act of writing down software requirement in a well-designed format organizes information, places borders around, solidfiers ideas and help break down the problems into its components parts in an orderly fashion.

It serves as an input to the design specification. As mentioned previously, the **SRS** serves as the parent document to subsequent document, such as the software design specification and statement of work .therefore, the **SRS** must contain sufficient detail in the functional system requirements so that a design solution can be devised.

It serves as a product validation check the **SRS** also serves as the parent document for testing and validation strategies that will be applied to the requirements for verification.

3.2 Functional Requirements:

The main purpose of functional requirement is to define all the activities or operations that take places in the system .These are derived through interactions with the users of the system .since requirements specifications is a comprehensive document and contains a lots of data ,it has been broken down in to different stages in the reports .

3.3 Non Functional Requirements:

Reliability:

Reliability is the correlation of an item, scale, or instrument with a hypothetical one, which truly measures what it is supposed to .Since the true instrument, is not available. The program according to the requirement can perform the intended functions. Error-handling-exception occurring while database needed to be addressed.

Usability:

Usability refers to the capability of the product to be understood, learned, and used and user friendly to users, when used under specified conditions. This section should include all of those requirements that affect usability.

Maintainability:

It is the ease with which a program/specification can be corrected if an error occur design a chain in requirement .specify attributes of software that relate to the ease of maintenance of the software itself .

Performance:

Performance is measured in terms of the output provided by the application. Requirement specifications play an important part in the analysis of a system .Only when the requirements specifications are properly given, it is possible to design a system, which will fill in to the required environment.

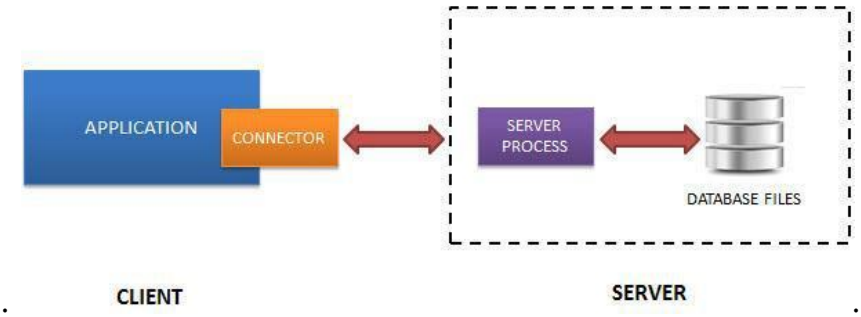
3.4 Technology Used

SQLITE

SQLite is a software library that provides a relational database management system. The lite in SQLite means light weight in terms of setup, database administration, and required resource.

SQLite has the following noticeable features: self-contained, serverless, zero-configuration, transactional.

Normally, an RDBMS such as MySQL, PostgreSQL, etc., requires a separate server process to operate. The applications that want to access the database server use TCP/IP protocol to send and receive requests. This is called client/server architecture.



3.4.2 Front End: JAVA AND XML

JAVA

Java is a popular general-purpose programming language and computing platform. It is fast, reliable, and secure. According to Oracle, the company that owns Java, Java runs on 3 billion devices worldwide..

Setting up your Java development environment

Install the JDK

Follow these steps to download and install the JDK:

Browse to Java SE Downloads and click the **Java Platform (JDK)** box to display the download page for the latest version of the JDK.

Agree to the license terms for the version you want to download.

Choose the download that matches your operating system and chip architecture.

Windows

Save the file to your hard drive when prompted.

When the download is complete, run the install program. Install the JDK to your hard drive in an easy-to-remember location such as C:\home\Java\jdk1.8.0_92. (As in this example, it's a good idea to encode the update number in the name of the install directory that you choose.)

XML

Xml (eXtensible Markup Language) is a mark up language.XML is designed to store and transport data.Xml was released in late 90's. it was created to provide an easy to use and store self describing data.XML became a W3C Recommendation on February 10, 1998.XML is not a replacement for HTML.XML is designed to be self-descriptive.XML is designed to carry data, not to display data.XML tags are not predefined.

4. ANALYSIS AND SYSTEM DESIGN

Existing System

In an existing system user need to go the tollgate directly and stand in long queues for hours together to pass the tollgate which is time consuming.

Users has to only pay by cash since there is no online payment .

Proposed System

The new system is totally computerized system.

A new system provides features like time efficiency to provide user profiles.

Reliable software application for managing tollgate management.

Online payment method is available in the system.

4.1 System Design

A computer procedure is a series of operations designed to manipulate data to produce outputs from a computer system. The procedure may be a single program or a series of programs. The details design of the computer procedure follow acceptance by a management of an outline design proposal .the aim now is to design procedure at lower level of details, which will define the detailed steps to be taken to produce the specified computer output .when complete, these procedure definitions together with data specifications are organized for programmers from which the required programs can be written.

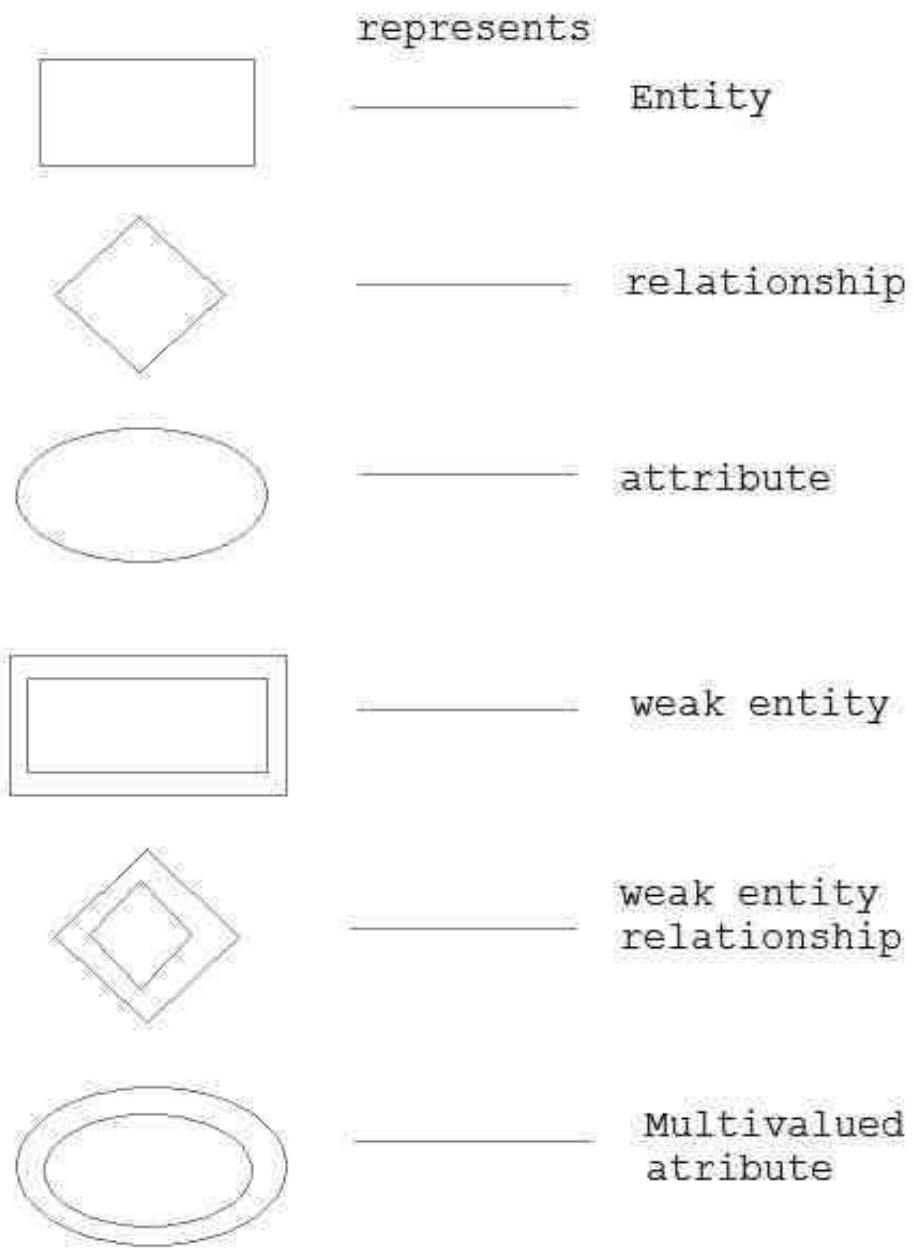
4.2 Data Dictionary

After carefully understanding the requirements of the requirements of the client the entire data storage requirements are divided into table. The below tables are normalized to avoid any anomalies during the course of data entry.

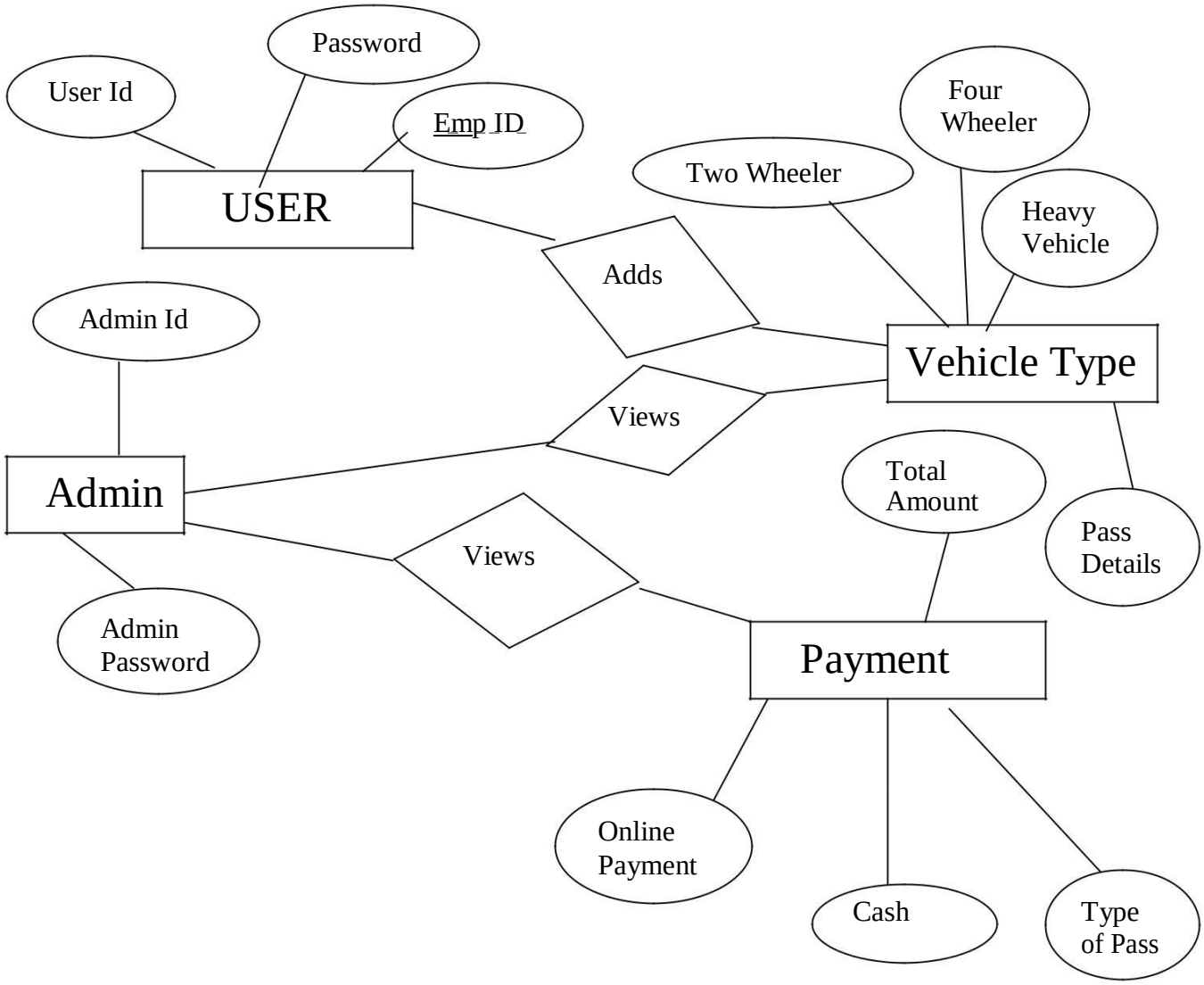
4.3 ER Diagram:

"E-R diagram are used to organize data as a relation, normalizing relations and finally obtaining a relational database model".

4.3.1 Symbols



4.4 ER Diagram :

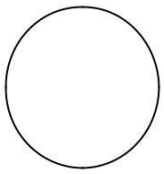


4.5 Data Flow Diagram

A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. It's a structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc. Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

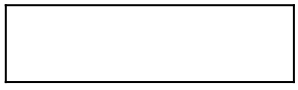
4.5.1 Data Flow Diagram Symbols: -

Process



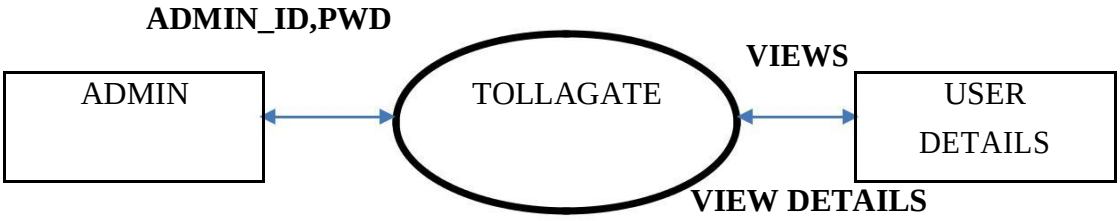
- **Activity**

- **Originator or Consumer of data**

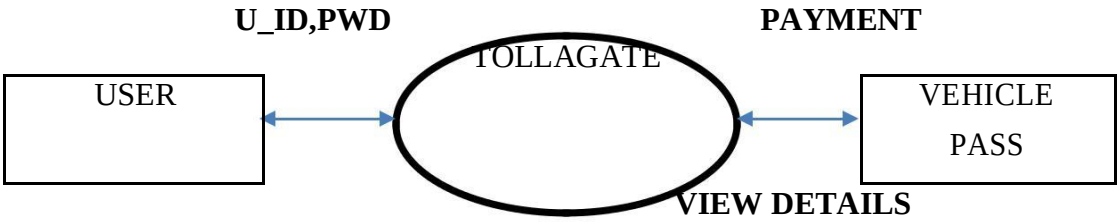


4.5.2 Data Flow Diagram

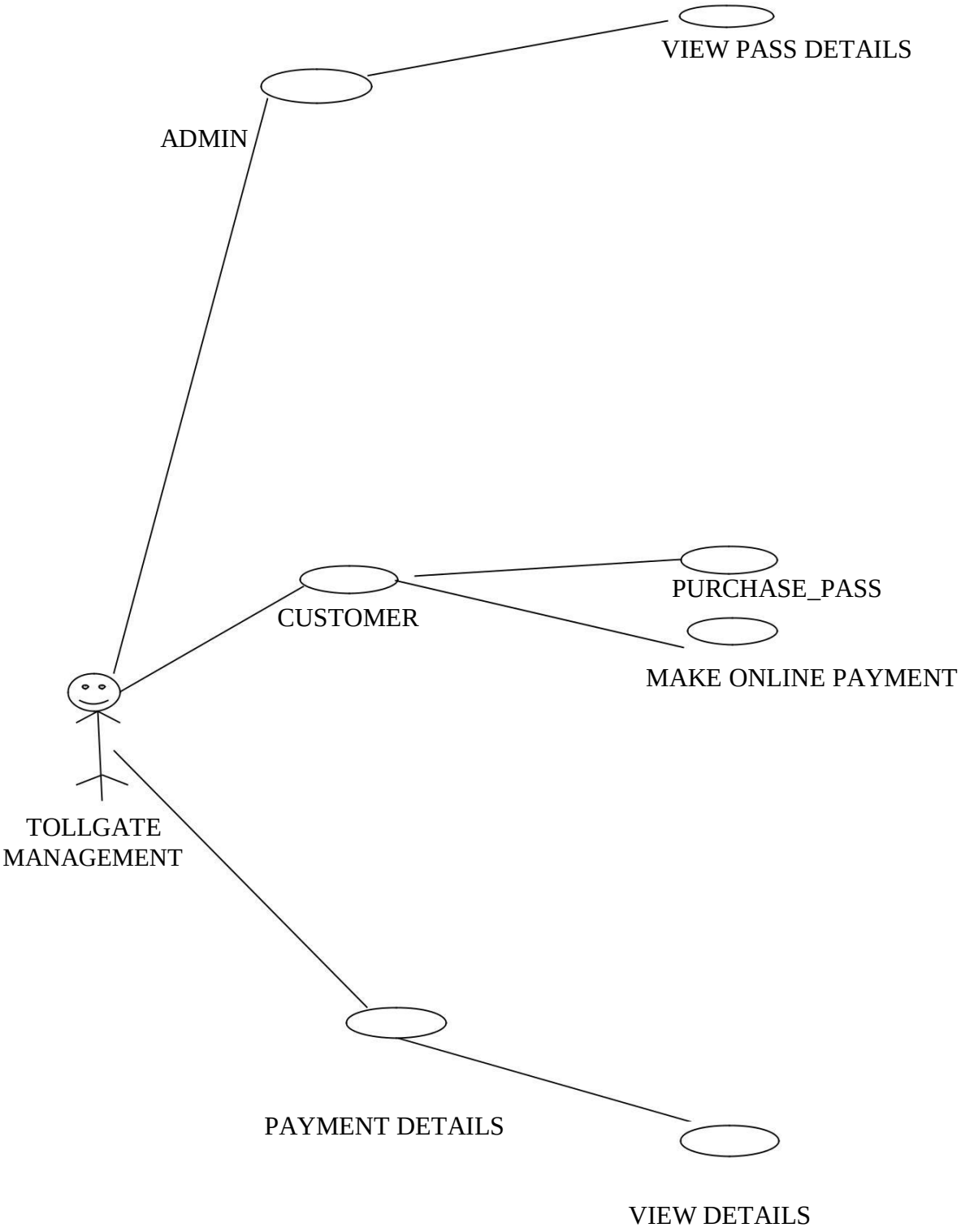
LEVEL 0



LEVEL 2



4.6 Use-Case Diagram



5. IMPLEMENTATION

The final and important phase in the system life cycle is the implementation of the new system. The term implementation has different meaning, ranging from the conventions of the basic applications to a complete replacement of the computer system. The procedure however is virtually the same. Implementation includes all those activities that take place to convert from old system to new. The coding of the proposed system is done with enough documentation. The method of implementation and the time scale to be adopted is found out initially. Next the system is tested properly and at the same time the user is trained with the new procedure. Proper implementation is essential to provide a reliable system to meet organization's requirement. Successful implementation may not guarantee improvement in the organization using the system, but it will prevent improper installation.

Table Design

User login

This table is used to store information about the Items.

FIELD NAME	DATA TYPE	SIZE	DESCRIPTION
U_id	NUMBER	10	User id
Name	VARCHAR	50	Name of user
Pass	VARCHAR	50	Password

Admin table:

This table is used to maintain the database of Login of Admin and user

FIELD NAME	DATA TYPE	SIZE	DESCRIPTION
Username	VARCHAR	50	Name of User
Password	VARCHAR	50	Password of the user

Pass Details :

This table is used to by the admin to view the feedback of user

FIELD NAME	DATA TYPE	SIZE	DESCRIPTION
Name	CHAR	50	Name of customer
Vehicle Number	VARCHAR	20	Vehicle Number
Vehicle Type	VARCHAR	30	Type of Vehicle
Pass Type	VARCHAR	11	Type of Pass
Starting Date	VARCHAR	30	Starting Date
Ending Date	NUMBER	10	Ending Date
Total Fares	NUMBER	10	Total Fares

Payment Table:

This table gives information about all products and brand.

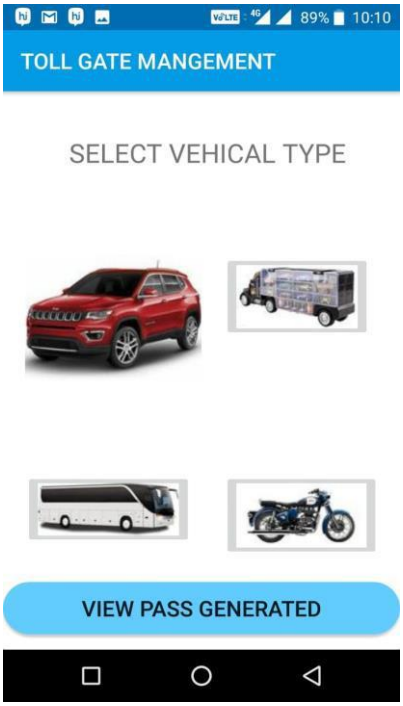
FIELD NAME	DATA TYPE	SIZE	DESCRIPTION
Product id	INT	10	Product id
Product name	CHAR	50	Name of Product
Brief description	Text	50	Brand name
Picture	CHAR	50	Small of Product
Big picture	CHAR	11	Big of Product
Description	Text	50	Description of product

5.1 SCREENSHOT

LOGIN PAGE

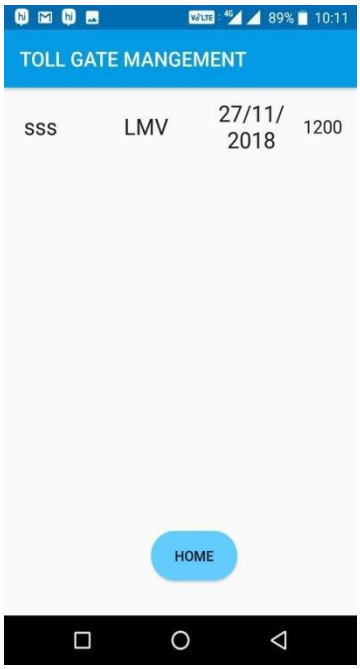
Description: This is a login page of the tollgate management. In the page user or admin can sign in.

VEHICAL TYPE PAGE



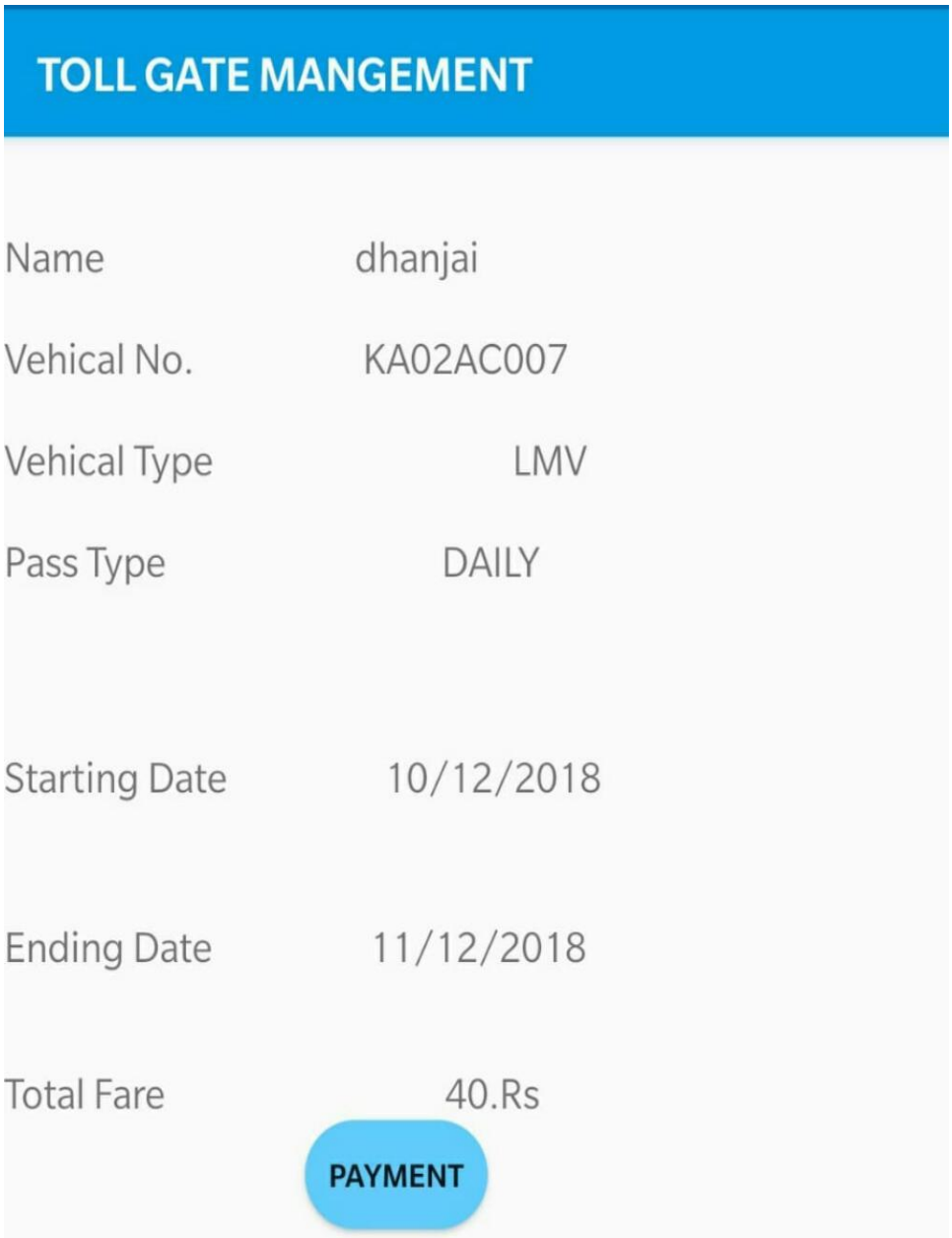
Description: This is a vehicle type page of the Tollgate Management application. In the page you can see the vehicle type.

ADMIN PAGE



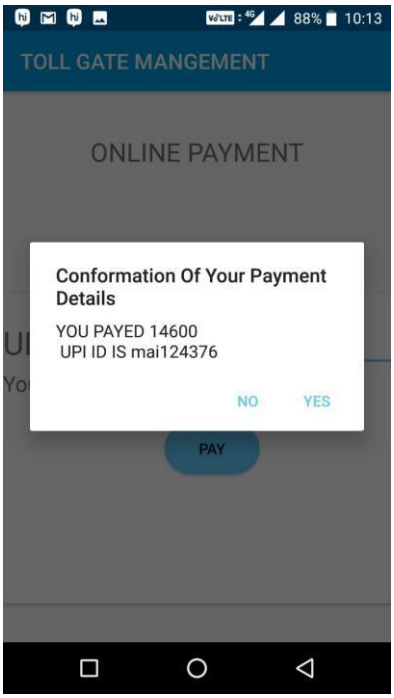
Description: This is a home page for admin, when he logs in which displays the details of user who has purchased the pass.

VIEW PASS DETAILS PAGE



Description: This is a home page for viewing the details of pass.

ONLINE PAYMENT PAGE



Description: This is a home page for online payment,which views the details of online payment.

5.2 CODING

MainActivity.java

```
package com.example.root.raveena_sanjana;
import android.content.Intent;
import android.os.Parcelable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle; import android.view.View;

import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.example.root.raveena.DatabaseHelper;
import java.io.Serializable;
public class MainActivity extends AppCompatActivity
 {
 DatabaseHelper  myDB;
Button register,login,detail,admin;
EditText user,pass;
@Override
protected void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
myDB = new DatabaseHelper(this);
login = (Button)findViewById(R.id.btnlogin);
detail=(Button)findViewById(R.id.button3); register
= (Button)findViewById(R.id.button2); user =
(EditText)findViewById(R.id.editText0); pass =
(EditText)findViewById(R.id.editText1); admin
=(Button)findViewById(R.id.admin); String us =
user.getText().toString();
```

```
String ps = pass.getText().toString();
Intent i1 = new Intent();
register.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v)
{
Intent intent = new Intent(MainActivity.this,register.class);
startActivity(intent);
        }
}
);
detail.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v)
{
Intent intent = new Intent(MainActivity.this,detail.class);
startActivity(intent);
}
 });
login.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v)
{
String us = user.getText().toString();
if (us.isEmpty() || ps.isEmpty())
{
 Toast.makeText(MainActivity.this,"ALL FIELAD ARE REQUIRED
",Toast.LENGTH_LONG).show(); }

 else
{
```

```
String isLogin = myDB.loginData(us,ps);
if (isLogin == "ok")
{
Toast.makeText(MainActivity.this,
"LOGIN SUCCESS ", Toast.LENGTH_LONG).show();
Intent intent = new Intent(MainActivity.this, vehical.class);
intent.putExtra("n",us);
startActivity(intent);
}
else
{
Toast.makeText(MainActivity.this, "LOGIN NOT SUCCESS ",
Toast.LENGTH_LONG).show(); }

}
}
});
admin.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v)
{
String us = user.getText().toString();
String ps = pass.getText().toString();
if (us.equals("admin") && ps.equals("admin"))
{
Intent intent = new Intent(MainActivity.this, admin.class);
startActivity(intent);
}
else
{
Toast.makeText(MainActivity.this,"NOT A ADMIN USER !!!!
",Toast.LENGTH_LONG).show();
```

```
}
}
}
);
}
}
```

Vehicledata.java

```
package com.example.root.raveena_sanjana;

import android.content.ContentValues;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase; import
android.support.v7.app.AppCompatActivity; import
android.os.Bundle; import android.view.View;

import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.example.root.raveena.DatabaseHelper; public
class vehicalDetail extends AppCompatActivity {
   DatabaseHelper sq;
   String n=null;
   String vn=null;
   String t=null;
   String pt=null;
   String sd=null;
   String ed=null;
   String to=null;
   int am;
   TextView t1,t2,t3,t4,t5,t6,t7;
```

```
Button pay;

@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_vehical_detail); sq =
   new DatabaseHelper(this);
   pay = (Button)findViewById(R.id.payment);
   Intent i = getIntent();
   n = i.getStringExtra("n");
   vn = i.getStringExtra("vn");
   t = i.getStringExtra("t");
   pt = i.getStringExtra("pt");
   sd = i.getStringExtra("sd");
   ed = i.getStringExtra("ed");
   to= i.getStringExtra("to");

   t1=(TextView)findViewById(R.id.name);
   t2 = (TextView)findViewById(R.id.vehno);
   t3= (TextView)findViewById(R.id.vehtype);
   t4= (TextView)findViewById(R.id.passtype);
   t5= (TextView)findViewById(R.id.startdate);
   t6= (TextView)findViewById(R.id.endDate);
   t7= (TextView)findViewById(R.id.totalFare);
   t1.setText(n);
   t2.setText(vn);
   t3.setText(t);
   t4.setText(pt);
   t5.setText(sd);
   t6.setText(ed);
   t7.setText(to+".Rs");

   pay.setOnClickListener(new View.OnClickListener() {
      @Override
```

```
        public void onClick(View v) {
           if(vn.isEmpty()){
              Toast.makeText(vehicalDetail.this,"Vehical Number can not be
EMPTY!!!!",Toast.LENGTH_LONG).show();
           }
           else {

              SQLiteDatabase db =sq.getWritableDatabase();
              ContentValues cv = new ContentValues();
              cv.put("NAME",n);
              cv.put("VEHNO",vn);
              cv.put("VEHTYPE",t);
              cv.put("PASS",pt);
              cv.put("STARTDATE",sd);
              cv.put("ENDDATE",ed);
              cv.put("FARE",to);
              Long res = db.insert("detail",null,cv);
              Toast.makeText(vehicalDetail.this,"
"+res,Toast.LENGTH_LONG).show();
              if(res == -1)
              {
                 Toast.makeText(vehicalDetail.this,"Data Not Inserted
",Toast.LENGTH_LONG).show();

              }
              else
              {
                 Toast.makeText(vehicalDetail.this,"Data Inserted
",Toast.LENGTH_LONG).show();
                 Cursor c = db.query("detail",new
String[]{"name"},null,null,null,null,null);

Toast.makeText(vehicalDetail.this,""+c.isFirst(),Toast.LENGTH_LONG).show(); Intent
                 i = new Intent(vehicalDetail.this, payment.class);
```

```
                i.putExtra("n", n);
                i.putExtra("to", to);
                i.putExtra("t", t);
                i.putExtra("pt", pt);
                i.putExtra("vn", vn);
                startActivity(i);
            }
          }
        }
    });
  }
}
```

Pass.java

```
package com.example.root.raveena_sanjana;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle; import android.view.View;

import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;

public class pass extends AppCompatActivity {
  String n=null;
  String r=null;
  String t=null;
  RadioButton r1;
  TextView v1;
  Button proceed;
  TextView tv;
```

```
  @Override

  protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_pass);
    final RadioGroup rg=(RadioGroup)findViewById(R.id.radioGroup1);
    final EditText e1=(EditText)findViewById(R.id.editText);
    tv =(TextView)findViewById(R.id.textView11);
    proceed =(Button)findViewById(R.id.button);
   Intent i = getIntent();
    t=i.getStringExtra("t");
    r = i.getStringExtra("a");
    n = i.getStringExtra("n");
    tv.setText(t);
    proceed.setOnClickListener(new View.OnClickListener()
{
       @Override

 public void onClick(View v)
{
          int selectedId = rg.getCheckedRadioButtonId(); r1 =
          (RadioButton) findViewById(selectedId) String
          vn=e1.getText().toString(); String
          vt=r1.getText().toString();
         Intent i1=new Intent(pass.this,pt.class);
          i1.putExtra("a",r);
          i1.putExtra("n", n);
          i1.putExtra("t", t);
          i1.putExtra("vn", vn);
          i1.putExtra("pt", vt);
          startActivity(i1);
       }
    });
```

```
    }
}
```

Passdetails.java

```
package com.example.root.raveena_sanjana;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class passDetail extends AppCompatActivity {

    DatabaseHelper sq;
    TextView t1,t2,t3,t4,t5,t6,t7;
   Button b1;
    String name=null;
    String n[];
    public int c1;
    @Override
    protected void onCreate(Bundle savedInstanceState)
{
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pass_detail);
```

```
sq = new DatabaseHelper(this);
t1=(TextView)findViewById(R.id.name2);
t2=(TextView)findViewById(R.id.vehno2);
t3=(TextView)findViewById(R.id.vehtype2);
t4=(TextView)findViewById(R.id.strDateData);
t5=(TextView)findViewById(R.id.endDateData);
t6 =(TextView)findViewById(R.id.finalViewAmount);
t7 =(TextView)findViewById(R.id.passtype2);
b1=(Button)findViewById(R.id.btnback);
sq.getReadableDatabase(); Intent i = getIntent();

final Context context = this;
name =i.getStringExtra("n");
SQLiteDatabase db =this.sq.getWritableDatabase(); String[] column = new
String[]{"name"};
Toast.makeText(passDetail.this,""+name,Toast.LENGTH_LONG).show();
b1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v)
{
      AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(context);
      alertDialogBuilder.setTitle("Conform go to BACK ");
      alertDialogBuilder
            .setMessage("IF YOU CLICK YES YOU CAN'T SEE THE DETAIL
AGAIN ")
            .setCancelable(false)
            .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
               public void onClick(DialogInterface dialog, int id) {
                  if this button is clicked, close
                  current activity
                  passDetail.this.finish();
                  Intent intent = new Intent(passDetail.this, vehical.class);
                  startActivity(intent);
```

```
                    }
                })
                .setNegativeButton("No", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id)
{
                        if this button is clicked, just close
                        the dialog box and do nothing
                        dialog.cancel();
                    }
                });

            AlertDialog alertDialog = alertDialogBuilder.create();

            show it
            alertDialog.show();
          }
       });
       getData();
   }
   public Cursor getData()
   {
       SQLiteDatabase db = this.sq.getReadableDatabase();
       Cursor c = db.rawQuery("select * from detail where NAME =?",new
String[]{name});
       while (c.moveToNext())
       {
          t1.setText(c.getString(0));
          t2.setText(c.getString(1));
          t3.setText(c.getString(2));
          t7.setText(c.getString(3));
          t5.setText(c.getString(4));
          t6.setText("YOU PAID "+c.getString(6)+"RS");
          t4.setText(c.getString(5));
```

```
        }
        if (c.isLast()){
            Toast.makeText(passDetail.this,"NO RECORD OF THE
USER",Toast.LENGTH_LONG).show();
            Intent intent = new Intent(passDetail.this,vehical.class);
            startActivity(intent);
        }
        return c;
    }
}
```

Cash.java

```
package com.example.root.raveena_sanjana;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle; import android.view.View;

import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;

public class pass extends AppCompatActivity {
    String n=null;
    String r=null;
    String t=null;
    RadioButton r1;
    TextView v1;
    Button proceed;
    TextView tv;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pass);
        final RadioGroup rg=(RadioGroup)findViewById(R.id.radioGroup1);
        final EditText e1=(EditText)findViewById(R.id.editText);
        tv =(TextView)findViewById(R.id.textView11);
        proceed =(Button)findViewById(R.id.button);

        Intent i = getIntent();
        t=i.getStringExtra("t");
        r = i.getStringExtra("a");
        n = i.getStringExtra("n");
        tv.setText(t);

        proceed.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int selectedId = rg.getCheckedRadioButtonId();

                r1 = (RadioButton) findViewById(selectedId);

                 String vn=e1.getText().toString();
                 String vt=r1.getText().toString();

                 Intent i1=new Intent(pass.this,pt.class);
                 i1.putExtra("a",r);
                 i1.putExtra("n", n);
                 i1.putExtra("t", t);
                 i1.putExtra("vn", vn);
                 i1.putExtra("pt", vt);
                 startActivity(i1);
            }
        });
    }
}
```

6. TESTING

No system design is ever perfect. A system is tested for online response, volume of transaction, stress, recovery from failure and usability. System testing requires a test plan that consists of several key activities and steps for programs, string, and systemanduser acceptance testing. Testing is the process to check the system to find out any error that many cause the system some trouble or hamper the integrity of the data.

6.1 Preparation of the Test Plan

Software testing is the process of executing a program with the intent of finding an error. A good test has a high probability of finding a yet undiscovered error. The test data was prepared keeping in mind the user requirement and expectation.

6.2 Method Used To Test Data

Test data was keened in through the data entry screen in the database.

Input/output were carried to ensure proper functioning of the system. It was confirmed that all the required validation and check are performed by giving invalid data input.

Testing for suitable error message/messages was done in case of invalid data entry.

Testing for proper report layout.

6.3 Testing Method

6.3.1 Unit Testing*:*

This type of testing involve individual testing of programs with respect to the desired expected output proper care is taken when entering the data checks were made to see that the data entered is correct and not of the specified bond(s), proper error message are flashed when an error occurs.

6.3.2 System Testing*:*

In this phase the system as a whole is tested. This testing phase will check the integrity of the data while it was transferred from one process to another. The interaction between the modules was checked to see whether a module procedure the required output, which would be needed as input of the another one.

6.3.3 Layout Testing*:*

This phase included the testing of the various form that are in the system check will be done to see that the data entries in the boxes provided the size of the boxes is appropriate. The various button places on the forms are functioning properly.

6.3.4 Period Validation Testing (For Report Generation):

This phase of testing was involved for the report preparation. This testing was done to see that the report prepare for the particular period are correctly generated and they are within the given data span and the date bonds mentioned are rightly followed.

6.3.5 Online Testing:

In online testing the system is handed over to user and allow working on it. If user finds any problem related to system. He informs to developer to correct the errors or problems.

Implementation:

This is crucial phase of system development .It involves introduction of new system into operation. This involves creating computer awareness on to job, training, installation of hardware, terminal if does not exist, client communication equipment before the system is up and made operational. User manuals are prepared and user is trained in its use.

Before implementation system was working manually. Few people were developed to run the computerized system. Data was fed at regular interval and monitories on basis of report and output. All the possible errors were recorded rectify and rested. The system after proper security and satisfaction was stepped in to next phase employees were given adequate training, about how to handle the system.

6.4 Tests & Results

Module tested	Test cases	Response(expected or not)	Action taken if it is not an expected response	Error fixed or not
Login module	(Username!=db Content) Input string	Invalid user	Re-enter	Y
Vehicle Type	Select vehicle details	SQL error	Re-enter	Y
Vehicle Number	Add the vehicle number	Successfully added	added	Y
Admin Module	Add username and Password	SQL error	deleted	Y
Email	Email Validation	Check Symbols	Re-enter	Y

7. CONCLUSION

The TOLLGATE MANAGEMENT, mobile application is a software which facilitates the people with safety and security.

This mobile application is very helpful to anyone who are driving through their vehicle, here the user can take pass before going to tollgate. This app facilitates to make exploring easy and brings all the necessary pass details .On login with clear view and attractive user interface. Only users can use this app. This is very useful application mainly for people who are driving through tollgate. When we feel that we are in emergency, for example travelling in the vehicle at peek time we can use this application, so that on one click we can buy the pass. So, there is no need of standing in long queues for hours together which is very time consuming and so this application is used which is less time consuming when compared to that of the existing method.

8. FUTURE ENHANCEMENT

Since we are developing a mini project in a given time duration, it is very difficult to provide all features in the first version itself .so the following are the few enhancement which can be implemented in further versions .To make exploring easy category has been provided for gadgets

Digital driver license will be provided in the application based on the information given by the driver and by providing the original license.

Vehicle number and digital license will be linked so that other people cannot do forgery.

BIBLIOGRAPHY

References

1. *Kamal Acharya. School management system project report. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254873.34023165/v1*
2. *Kamal Acharya. A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254873.30191075/v1*
3. *Kamal Acharya. A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT. Authorea. August 01, 2024 DOI: https://doi.org/10.22541/au.172254872.26972790/v1*
4. *Kamal Acharya. Web chatting application project report management system. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254871.18588592/v1*
5. *Kamal Acharya. RETAIL STORE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254871.14590154/v1*
6. *Kamal Acharya. SUPERMARKET MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252491.19145062/v1*
7. *Kamal Acharya. SOCIAL MEDIA MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252491.11210579/v1*
8. *Kamal Acharya. Online music portal management system project report. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252488.89734698/v1*
9. *Kamal Acharya. COLLEGE BUS MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172245277.70798942/v1*
10. *Kamal Acharya. AUTOMOBILE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172245276.67982593/v1*
11. *Kamal Acharya. Ludo management system project report. Authorea. July 31, 2024 DOI: https://doi.org/10.22541/au.172243999.98091616/v1*
12. *Kamal Acharya. Literature online quiz system project report. Authorea. July 31, 2024 DOI: https://doi.org/10.22541/au.172243825.53562953/v1*
13. *Kamal Acharya. Avoid waste management system project. Authorea. July 29, 2024 DOI: https://doi.org/10.22541/au.172228528.85022205/v1*
14. *Kamal Acharya. CHAT APPLICATION THROUGH CLIENT SERVER MANAGEMENT SYSTEM PROJECT. Authorea. July 29, 2024. DOI: https://doi.org/10.22541/au.172228527.74316529/v1*
15. *Acharya, K. (2024). online taxi booking management system project report. DOI: https://doi.org/10.5281/zenodo.13642990*
16. *Acharya, K. (2024). Chat application through client server management system project report. DOI: https://doi.org/10.5281/zenodo.13642971*
17. *Acharya, K. (2024). online blood donation management system project report. DOI: https://doi.org/10.5281/zenodo.13642922*
18. *Acharya, K. (2024). online course register system project report. DOI: https://doi.org/10.5281/zenodo.13642791*
19. *Acharya, K. (2024). Fruit shop management system project report. DOI: https://doi.org/10.5281/zenodo.13642698*

20. *Acharya, K. (2024). Toll tax management system project report. DOI: https://doi.org/10.5281/zenodo.13642635*
21. *Acharya, K. (2024). Health insurance claim management system project report. DOI: https://doi.org/10.5281/zenodo.13642551*
22. *Acharya, K. (2024). A case study of cinema management system project report. DOI: https://doi.org/10.5281/zenodo.13642433*
23. *Acharya, K. (2024). Lundry management system project report. DOI: https://doi.org/10.5281/zenodo.13642310*
24. *Acharya, K. (2024). Student information management system project report ii. DOI: https://doi.org/10.5281/zenodo.13642142*
25. *Acharya, K. (2024). A case study of online ticket booking system project report. DOI: https://doi.org/10.5281/zenodo.13641983*
26. *Acharya, K. (2024). Training and placement cell management system. DOI: https://doi.org/10.5281/zenodo.13625502*
27. *Acharya, K. (2024). Graphical password management system project report. DOI: https://doi.org/10.5281/zenodo.13625484*
28. *Acharya, K. (2024). Airplane game management system project report. Zenodo. DOI: https://doi.org/10.5281/zenodo.13372675*
29. *Acharya, K. (2023). Room finder management system project report. DOI: https://doi.org/10.5281/zenodo.13372710*
30. *Acharya, K. (2024). Bouncing ball content management system project report. DOI: https://doi.org/10.5281/zenodo.13374350*
31. *Acharya, K. (2024). Clock report management system project report. DOI: https://doi.org/10.5281/zenodo.13374306*
32. *Acharya, K. (2024). Tictactoe game management system project report. DOI: https://doi.org/10.5281/zenodo.13374244*
33. *Acharya, K. (2024). Company visitor management system projec report. DOI: https://doi.org/10.5281/zenodo.13372931*
34. *Acharya, K. (2024). A file system for mobile computing project report. DOI: https://doi.org/10.5281/zenodo.13372919*
35. *Acharya, K. (2024). Clinic management system project. DOI: https://doi.org/10.5281/zenodo.13372899*
36. *Acharya, K. (2024). Clothes store management system project report. DOI: https://doi.org/10.5281/zenodo.13372884*
37. *Acharya, K. (2024). Student feedback management system project report. DOI: https://doi.org/10.5281/zenodo.13372872*
38. *Acharya, K. (2024). College training and placement system project report. DOI: https://doi.org/10.5281/zenodo.13372847*
39. *Acharya, K. (2024). Telephone billing system project report project. DOI: https://doi.org/10.5281/zenodo.13372841*
40. *Acharya, K. (2023). Soccer management system report project. DOI: https://doi.org/10.5281/zenodo.13372825*
41. *Acharya, K. (2022). Online auditorium booking system project report. DOI: https://doi.org/10.5281/zenodo.13372817*
42. *Acharya, K. (2022). Expense tracker management system project report. DOI: https://doi.org/10.5281/zenodo.13372807*
43. *Acharya, K. (2024). Flower shop billing management system project. DOI: https://doi.org/10.5281/zenodo.13372804*